Tipagem de três linguagens diferentes:

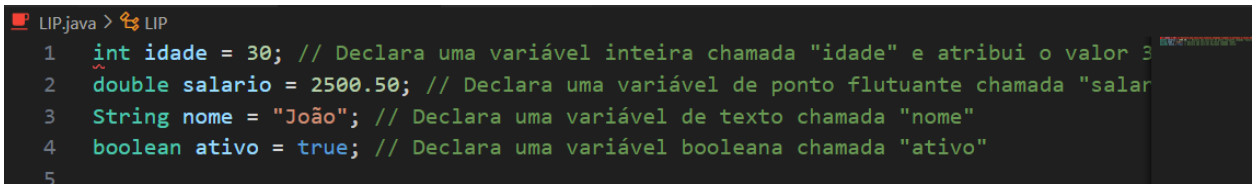
- JAVA
- Python
- C

É uma linguagem **fortemente tipada e estaticamente tipada**, o que significa que cada variável deve ter um tipo de dado ***declarado explicitamente***, e esse tipo é verificado durante a compilação.

-> Tipos Primitivos	-> Tipos de Referência
inteiro (byte, short, long)	Classes, interfaces e arrays
flutuantes (float, double)	
caracteres (char, string)	
booleanos (boolean)	

Tipagem Estática	Tipagem Forte
Cada variável é conhecida no momento da compilação, qualquer incompatibilidade de tipo é detectada na compilação, o que evita dar erro no tempo de execução.	A tipagem forte garante que operações entre tipos incompatíveis não sejam permitidos. Isso evita erros de execução relacionados a tipos.

Exemplo de declaração de tipo:



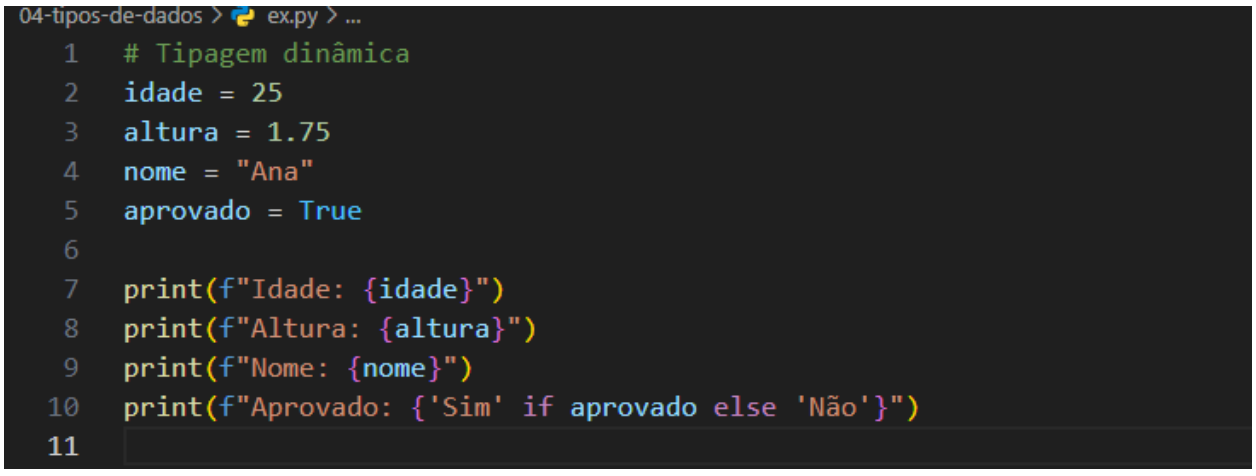
- Java
- PYTHON
- C

Python é uma linguagem de tipagem dinâmica e forte. Isso significa que o tipo de uma variável é determinado em tempo de execução (dinamicamente) e não é necessário declarar explicitamente o tipo da variável. Além disso, operações entre tipos incompatíveis não são permitidas, e resultam em erro.

Tipagem Dinâmica	Tipagem Forte
O tipo de uma variável não é fixo, ou seja, pode mudar ao longo do programa. O python faz a conclusão do tipo em tempo de execução, com base no valor de atribuição da variável.	Não permite conversas entre tipos de dados incompatíveis. Por exemplo, se tentar somar um número inteiro a uma string, o Python irá gerar um erro em vez de tentar fazer a conversão automaticamente.

-> Tipos primitivos	-> Tipos de Referência
`int` (em Python, não há `byte`, `short` ou `long`. `int` suporta números grandes nativamente)	list
`float`, `complex` (para números complexos)	tuple
`str` (substitui o `char` e `string`)	dict
`bool` (`True`, `False`)	Objetos/Classes

Exemplo de declaração de tipo:



- Java
- Python
- C

É uma linguagem estaticamente tipada e fracamente tipada. Ou seja, o tipo de cada variável deve ser declarado no momento da compilação, mas o compilador permite conversões (implícitas ou explícitas) entre tipos incompatíveis, o que pode levar a comportamentos inesperados em tempo de execução.

-> Tipos Primitivos	-> Tipos de Referência
inteiro (byte, short, long, long long)	Classes, interfaces e arrays
flutuantes (float, double, long double)	
caracteres (char)	
booleanos (boolean)	

> *C não possui **tipos de referência** como classes ou objetos nativamente. O ponteiro (`*`) é usado para acessar endereços de memória, sendo uma das principais formas de abstração.*

Tipagem Estática	Tipagem Fraca
Cada variável é conhecida no momento da compilação, qualquer incompatibilidade de tipo é detectada na compilação, o que evita dar erro no tempo de execução.	No entanto, o C permite conversões implícitas (coerção) entre tipos distintos, como entre `int` e `float`, ou `char` e `int`, o que caracteriza sua **tipagem fraca**. Isso pode causar erros silenciosos ou comportamentos indefinidos se não for bem controlado.

Exemplo de declaração de tipo:

04-tipos-de-dados > c.c

```c
#include <stdio.h>
#include <stdbool.h>

int main() {
    int idade = 25;
    float altura = 1.75;
    char inicial = 'A';
    bool aprovado = true;

    printf("Idade: %d\n", idade);
    printf("Altura: %.2f\n", altura);
    printf("Inicial: %c\n", inicial);
    printf("Aprovado: %s\n", aprovado ? "Sim" : "Não");

    return 0;
}
```